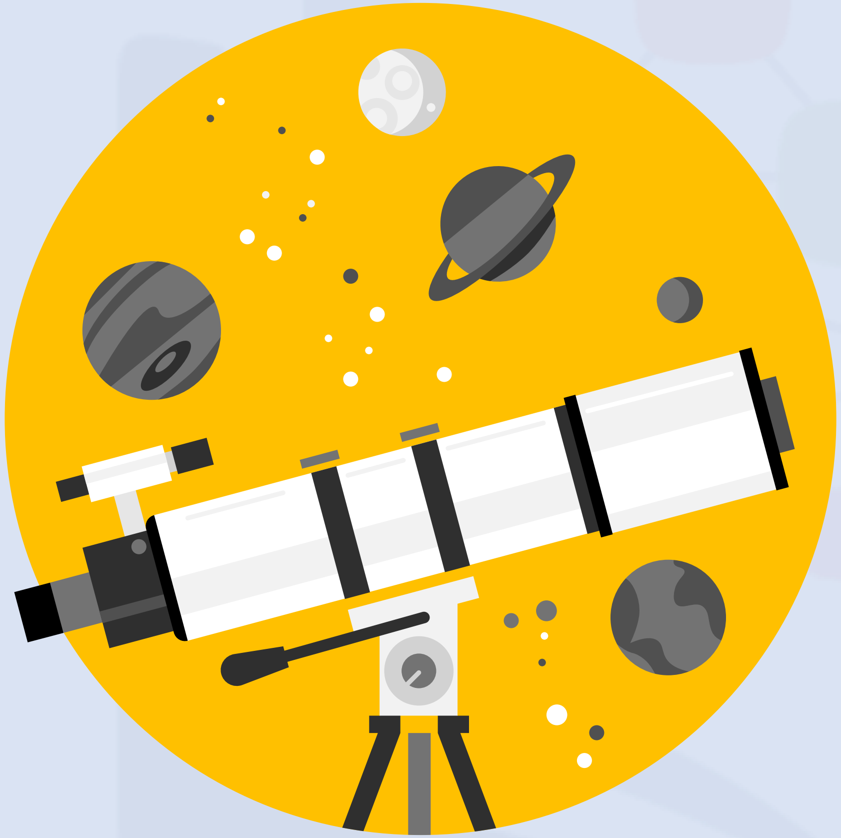
SpaceY :
Predicting Rocket Landing & Reuse

Majesam Hussain : 2024

IBM Developer

SKILLS NETWORK

OUTLINE



- Executive Summary
- Introduction
- Methodology
  - Data Collection & Wrangling
  - EDA with SQL & Visualisation
  - Mapping with Folium & Plotly Dash
  - Predictive Analysis
- Results
  - Visualization – Charts
  - Dashboard
- Conclusion
  - Findings & Implications
- Appendix

EXECUTIVE SUMMARY

The methodologies used were:

- Data Collection
- Data Wrangling
- Exploratory Data Analysis
  - With SQL
  - With Visualisation
- Mapping & Dashboards
  - Mapping with Folium
  - Dashboard with Plotly Dash
- Predictive Analysis using Machine Learning
- Findings and Implications

INTRODUCTION

Background

- The SpaceX rocket **Falcon9** is advertised to be more cost-efficient than other providers as it is reused in the first stage.
- In this project, we aim to predict if the first stage will land successfully.
- If we can determine if the first stage will land, then we can determine the cost of a launch.
- Some of the questions we need to answer:
  - What variables determine landing success?
  - What conditions are needed for the best landing?

METHODOLOGY

- Data was collected using SpaceX REST API, and historical launch records were web scraped from Wikipedia.
- For Data Wrangling, the data was prepared and outcomes for successful landings was determined.
- For Exploratory Data Analysis (EDA) , SQL and Visualisations were used to look at flights and launch sites and the relationships between variables (payload, orbit type). One-hot-encoding was used on the category features.
- This data was interactively visualised using Folim mapping and then displayed on a dashboard using Plotly Dash.
- Predictive Analysis was then performed using Machine-Learning and using the following models:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K Nearest Neighbour (KNN)

METHODOLOGY : Data Collection & Wrangling

Data Collection & Wrangling

- Data for the rocket launch data was retrieved from the SpaceX REST API using the GET request
- This was decoded using Json, then global variables and a dictionary were created for the key variables.
- API used to to get information about the launches
  - These included:
    - From Rocket > Booster Name
    - From Payload > Mass & Orbit
    - From Launchpad > Launch Site and coordinates
    - From Cores > Outcome of Landing (also includes details of # of flights, core reuse etc)
- To deal with missing values, data wrangling was performed using mean function, and the replace function.

Web Scraping

- To collect the Falcon 9 historical launch records, Wikipedia was used to look at the HTML table of the data.
- BeautifulSoup package was used to extract the table:
  - All columns and variable names extracted from the HTML table header
  - HTML table parsed and data frame created.

METHODOLOGY : EDA with SQL & Visualisation

SQL performed to gain information insight into the dataset

- The aim was to determine if the first stage will land then we can determine the cost of the launch.

- Therefore the following actions were performed using SQL:

    - DISTINCT launches displayed
    - Showing launch sites with 'CCA'
    - The total and average payload mass carried by boosters launched by NASA
    - Dates of successful landing outcomes and the list of successful/failure mission outcomes
    - Ranking the outcomes
    - Visualisations using seaborn to create scatter plots.
    - Line and Bar Graphs to show trends and categories.

METHODOLOGY : EDA with Folium & Plotly Dash

- Using **Folium**, each launch sites location was added using latitude and longitude coordinates
- A Circle marker for each location
- The launch outcomes for each sites was then added to see which sites have a high success rate with a GREEN marker if the launch was successful and a RED marker if the launch failed.
- Next, lines were added for the distances between a launch site to its proximities was calculated (with a marker icon for nearest railway, highway and city.

- An interactive dashboard was created with **Plotly Dash**
- In order to help with answering
  - Site with the largest successful launches
  - Site with the highest launch success rate
  - Payload range with highest launch success rate
  - Payload range with lowest launch success rate
  - F9 booster version with highest launch success rate
- Launch site Drop-down Input Component added
- Callback function to render site pie chart
- Range Slider for Payload
- Callback function render payload scatter plot

METHODOLOGY : Predictive Analysis

Machine Learning performed for prediction

- Created Numpy array and data standardised.
- Data split into train and test.
- Set Parameters using GridSearchCV,
- Used different machine learning algorithms to calculate accuracy of the models:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree
  - K Nearest Neighbour (KNN)
- Examined confusion matrix for each type to determine best method.

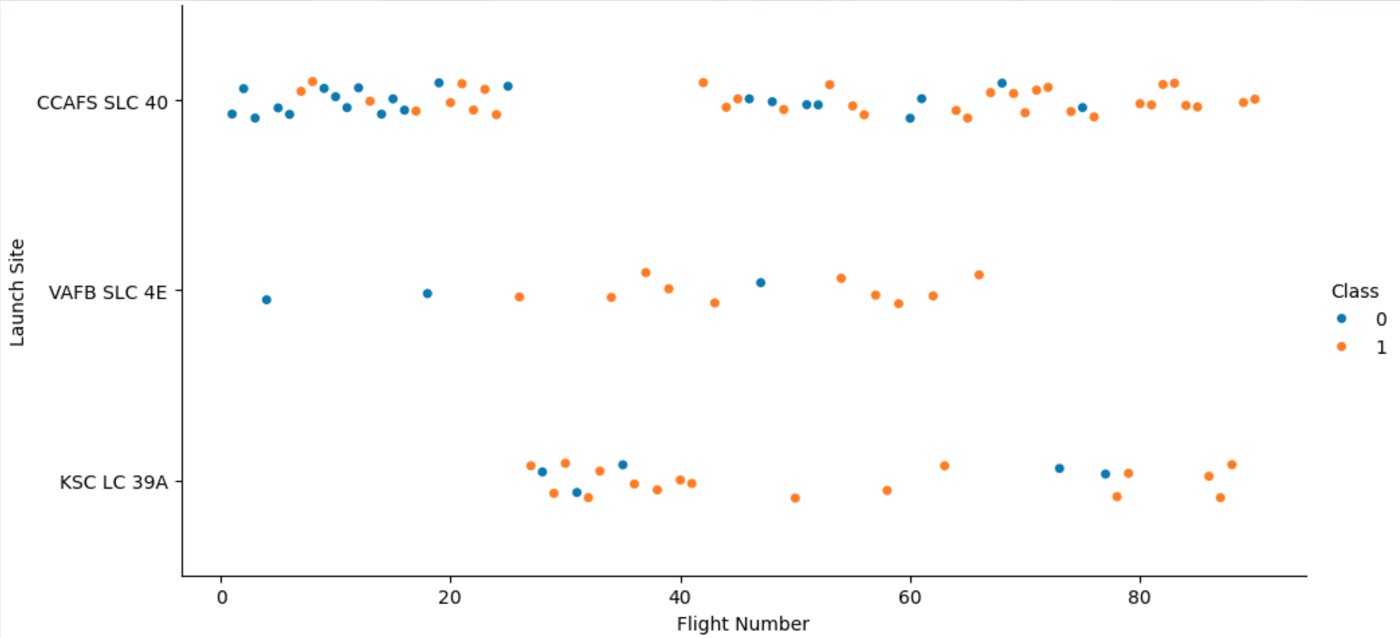
RESULTS

Summary

- EDA results; with SQL and Data Visualisation
- Mapping by Folium and Interactive Dashboard by Plotly Dash
- Predictive Analysis using Machine Learning Algorithms

RESULTS : EDA with SQL & Visualisation

- First we looked at the relationship between **Flight Number vs Launch Site**



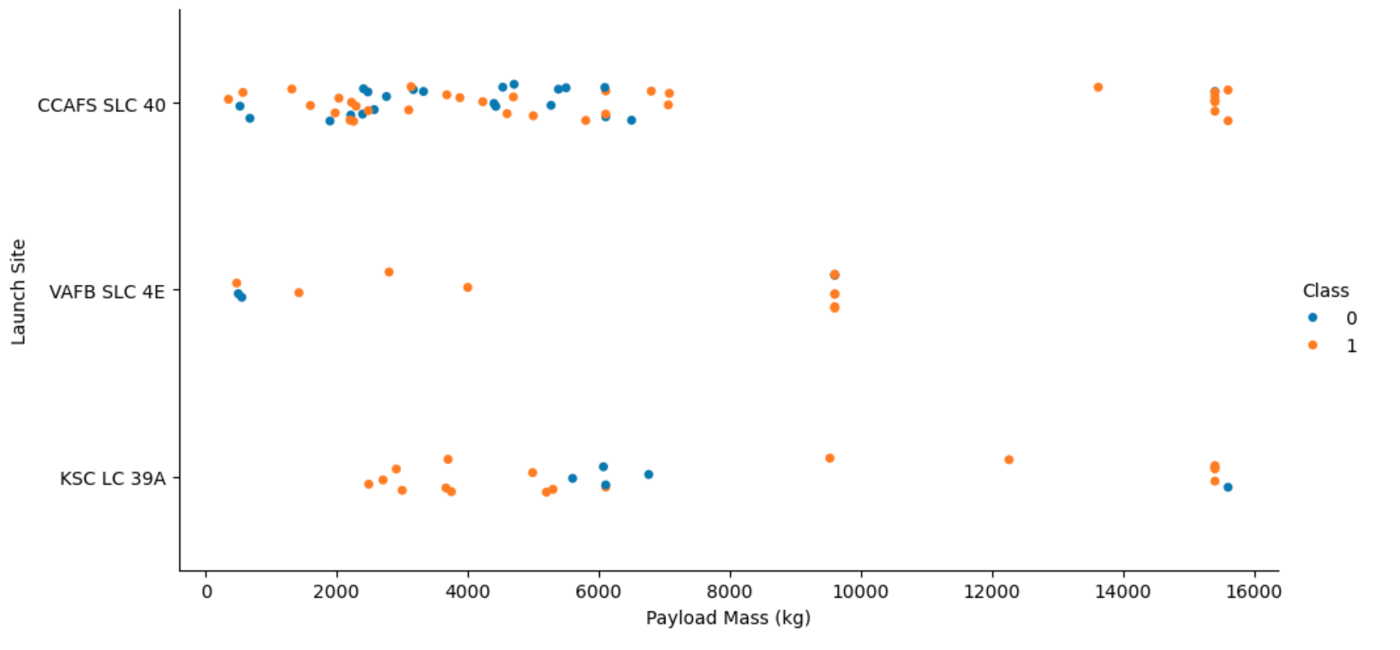
= success

= fail

- Lower flight numbers show lower success rate, whilst higher flight numbers show higher success rate

RESULTS : EDA with SQL & Visualisation

- Next, we looked at the relationship between **Payload vs Launch Site**



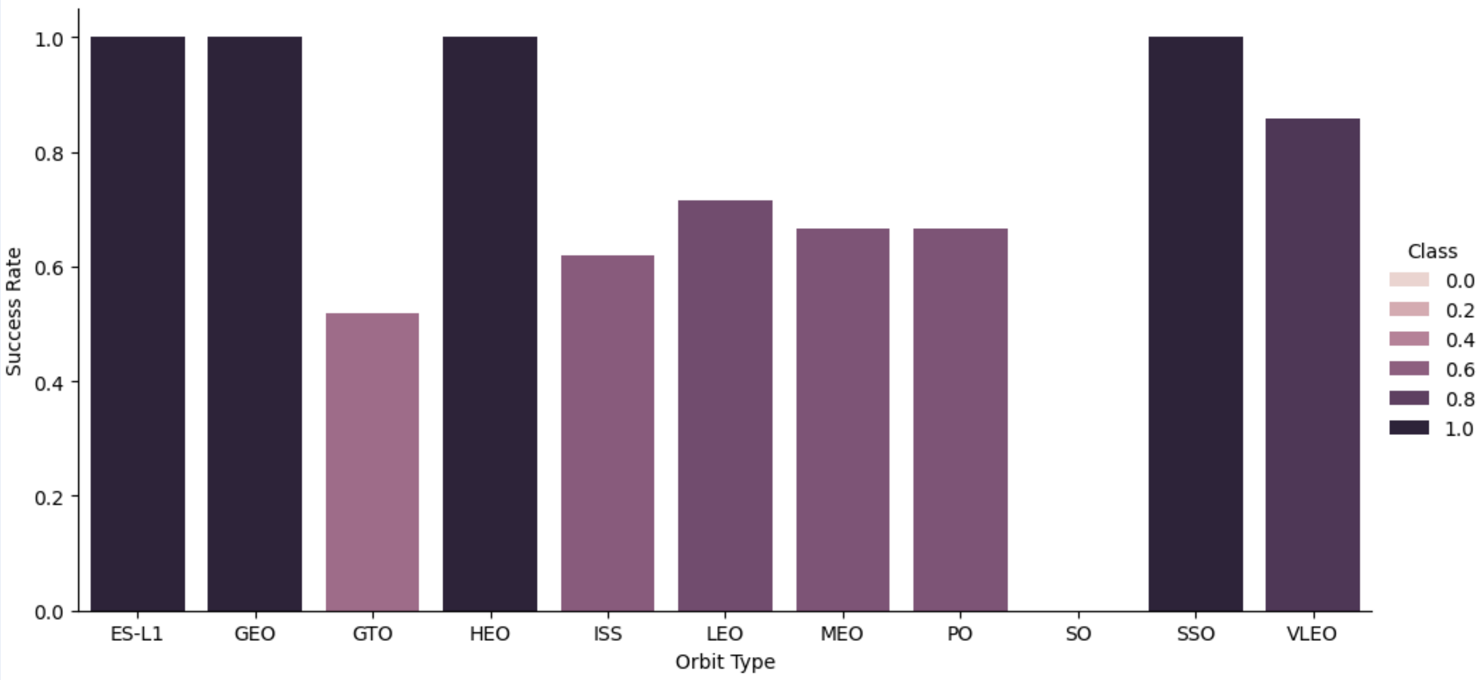
= success

= fail

- The greater the mass of the payload, the higher the success rate.

RESULTS : EDA with SQL & Visualisation

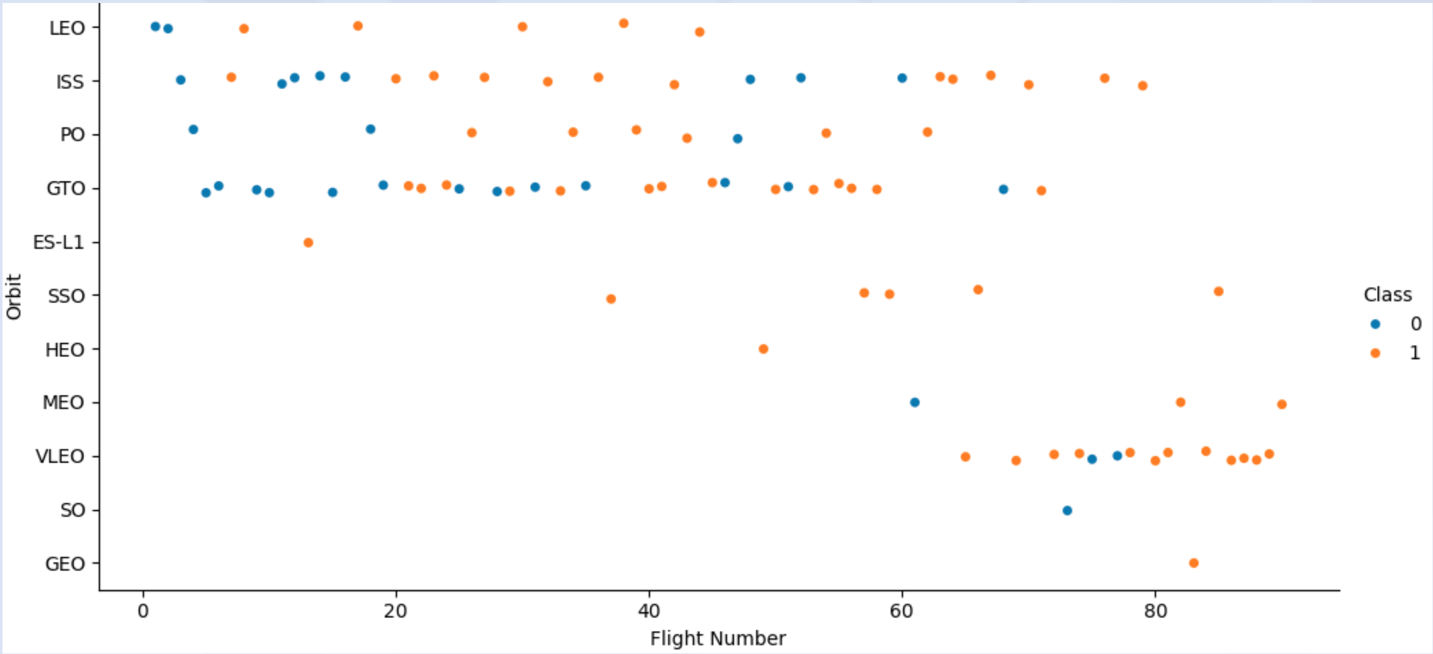
- We then visualised the relationship between **Success Rate of each orbit type.**



- The orbit types ES-L1, GEO, HEO and SSO have the highest success rate.
- Lowest success rate is for orbit type GTO

RESULTS : EDA with SQL & Visualisation

- Next, we looked at the relationship between **Flight Number vs Orbit type**



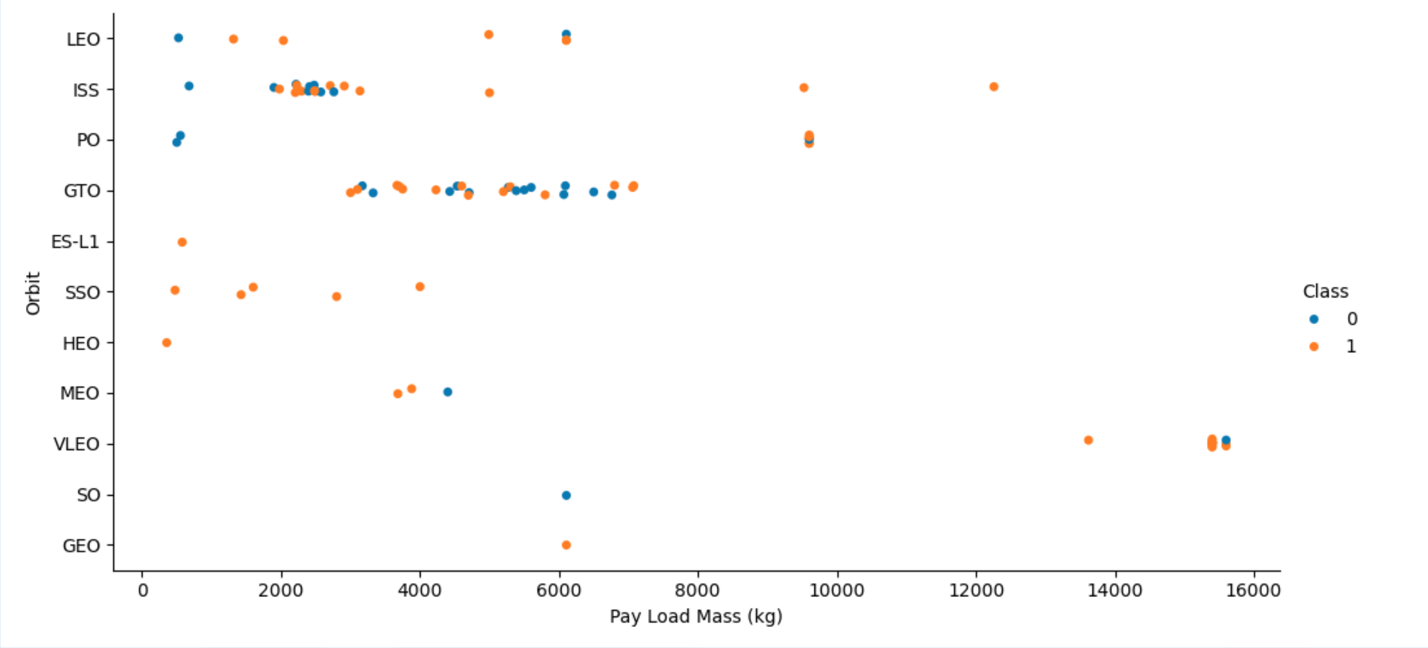
= success

= fail

- For LEO, the higher the flights the more likely successful. No other notable inferences made and GTO remains independent.

RESULTS : EDA with SQL & Visualisation

- Next, we looked at the relationship between **Payload vs Orbit type**



= success

= fail

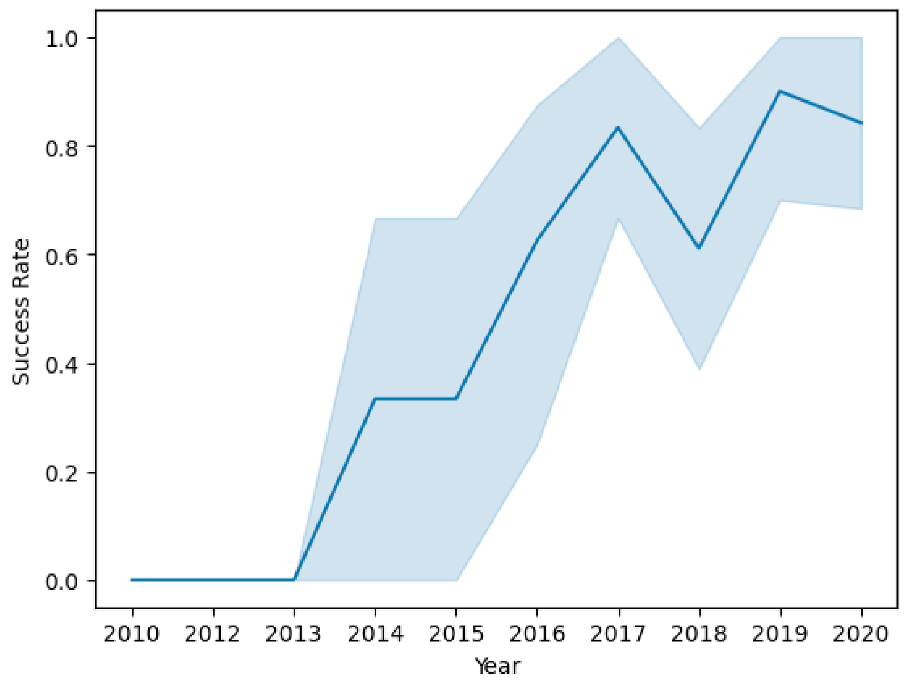
- For PO, LEO and ISS the heavier the payload, the more positive the landing rate. No other notable inferences made.

IBM Developer

SKILLS NETWORK

RESULTS : EDA with SQL & Visualisation

- We can also observe the **launch success yearly trend**



- Success Rate has kept increasing since 2013 (figures available until 2020).
- Notable dip in 2018.

RESULTS : EDA with SQL & Visualisation

- We also derived other key information from the dataset:
- Total Payload mass carried by booster launched by NASA, and average payload mass by Booster F9 v1.1

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS_NASA_CRS
FROM SPACEXTABLE
WHERE Customer= 'NASA (CRS)'

 * sqlite:///my_data1.db
Done.
```

TOTAL_PAYLOAD_MASS_NASA_CRS
45596

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS 'AVG_PAYLOAD_MASS_BOOSTER_F9_v1.1'
FROM SPACEXTABLE
WHERE Booster_Version LIKE 'F9 v1.1%'

 * sqlite:///my_data1.db
Done.
```

AVG_PAYLOAD_MASS_BOOSTER_F9_v1.1
2534.6666666666665

- The date of the first successful landing outcome

MIN(Date)
2015-12-22

- List of boosters which have success in drone ship and payload > 4000 and less <6000KG

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

RESULTS : EDA with SQL & Visualisation

- Total Number of successful and failure mission outcomes

```
%%sql
SELECT COUNT(Landing_Outcome) AS 'Outcomes'
FROM SPACEXTABLE
WHERE Landing_Outcome LIKE 'Success%' OR Landing_Outcome LIKE'Failure%'
```

* sqlite:///my_data1.db
Done.

Outcomes
71

- Names of booster versions which have carried the maximum payload mass

```
%%sql
SELECT Booster_Version from SPACEXTABLE
where PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

* sqlite:///my_data1.db
Done.

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

- The month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.

```
%%sql
SELECT SUBSTR(Date,6,2) AS 'Month', Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTABLE
WHERE SUBSTR(Date,0,5)='2015' AND Landing_Outcome='Failure (drone ship)'
```

* sqlite:///my_data1.db
Done.

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

RESULTS : EDA with SQL & Visualisation

- We then ranked the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

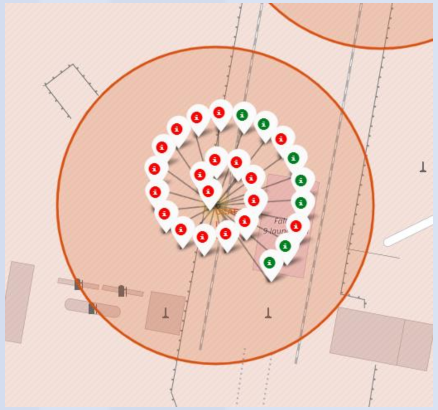
```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) AS Landing_Count
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY Landing_Count DESC
```

```
 * sqlite:///my_data1.db
Done.
```

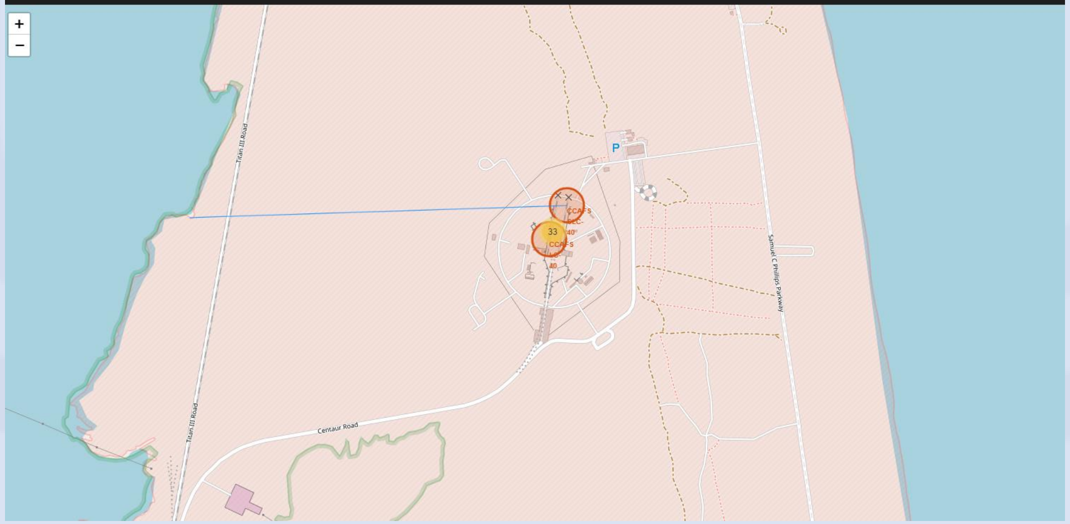
Landing_Outcome	Landing_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

RESULTS : Mapping with Folium

- Launch sites were marked on a map
- Latitude and Longitude were added for each site and map object was created with a center location of NASA Johnson Space Centre at Houston, Texas.
- Each success/failed launch was marked for the each launch site.
- Then the distances between a launch site to its proximites was calculated
  - For example; coastline



```
# Create a `folium.PolyLine` object using the coastline coordinates and launch site coordinate
coordinates = [[launch_site_lat,launch_site_lon],[coastline_lat,coastline_lon]]
lines=folium.PolyLine(locations=coordinates, weight=1)
site_map.add_child(lines)
```

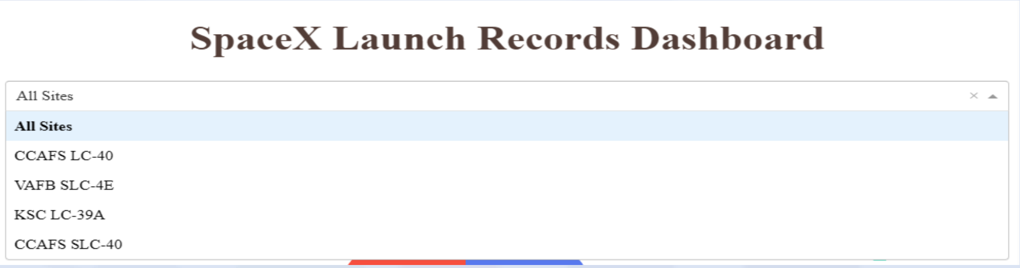


RESULTS : Mapping with Folium

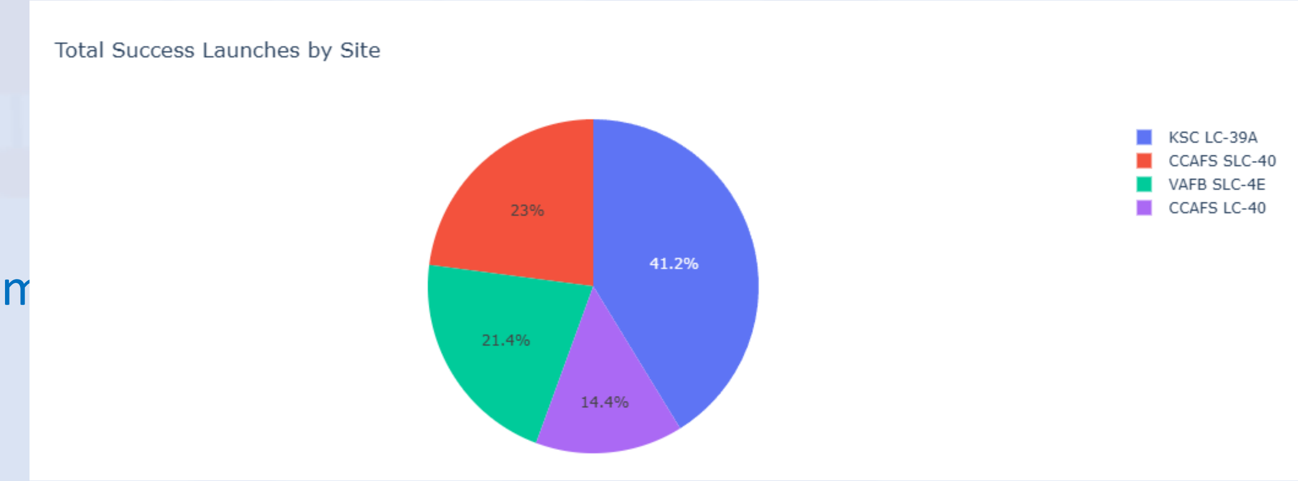
- Markers were then created to show distance to nearest city, railway line and highway.
- Launch sites are fairly close to railways to ensure good transport links for personnel and parts.
- Also, close to highways as that is the most common form of transport.
- The launch sites are close to the coast, which is important for safety and ensures unsuccessful launches are routed to the sea.
- Launch sites keep a sizeable distance away from cities to avoid densely populated areas, in the event of an unsuccessful launch, which can be a safety concern.

RESULTS : Dashboard with Plotly

- Dashboard created with Drop-down option for All Sites.



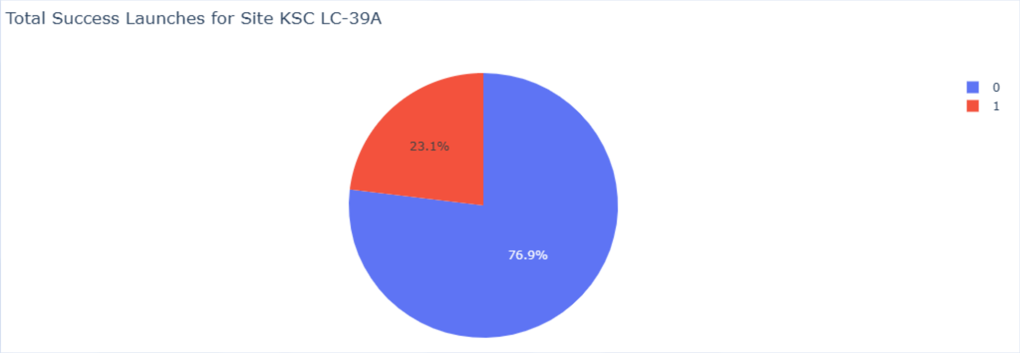
- The pie chart for Total Success Launches for All Sites



- KSC-LC-39A had the most successful launches from all sites (at 41.2%)

RESULTS : Dashboard with Plotly

- KSC-LC-39A achieved a largely successful launch success rate (at 76.9%).



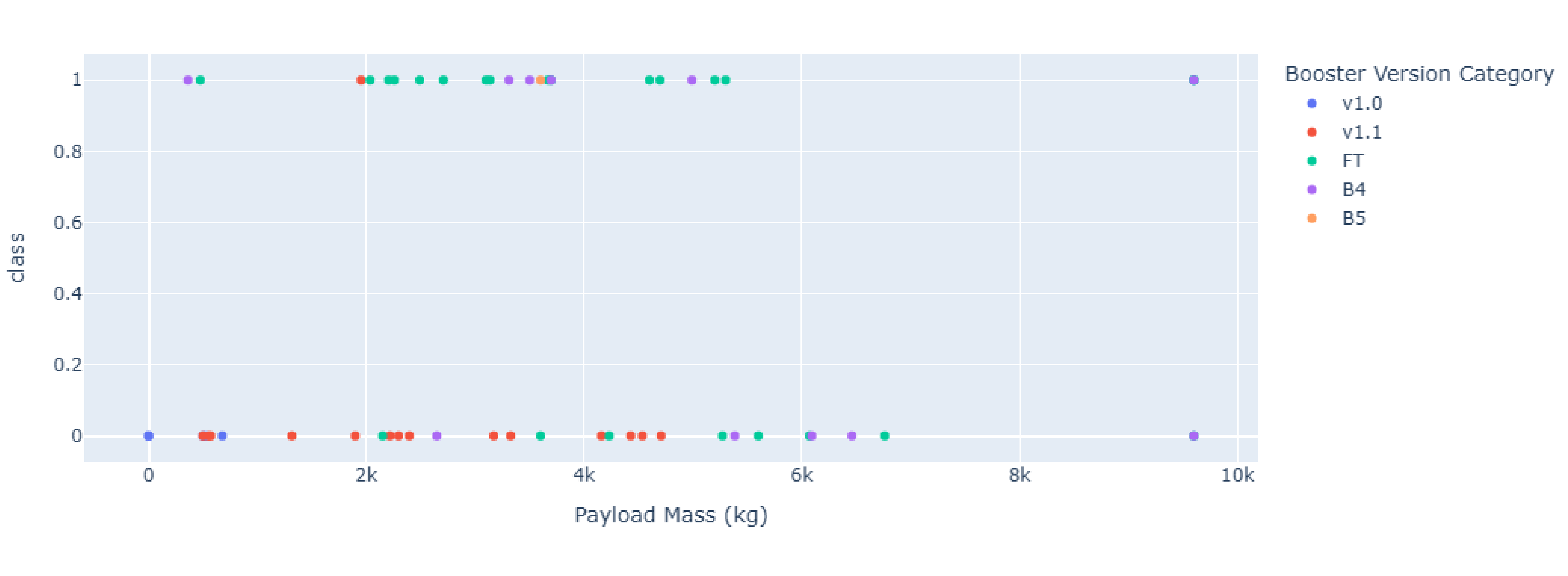
- A slider was added to look at payload range

IBM Developer

SKILLS NETWORK

RESULTS : Dashboard with Plotly

- The success rate for low weighted payload is higher than heavy weighted payload.

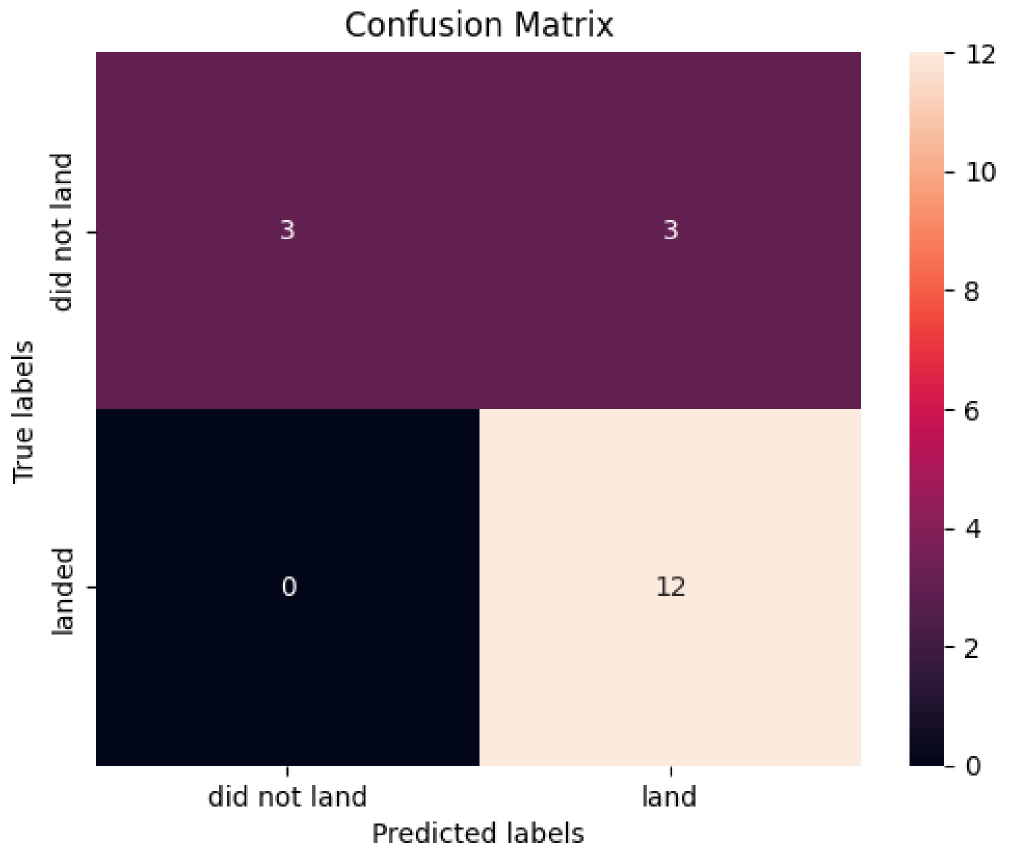


RESULTS : Predictive Analysis

The models were trained and hyperparameters selected using the function GridSearchCV

Accuracy

- To determine the best model, each score was calculated using the method 'score' and confusion matrix plotted; these were the same for all models
- The sample sizes may not have been suitable as scores were the same as were the confusion matrix
- High number of false positives for each model



RESULTS : Predictive Analysis

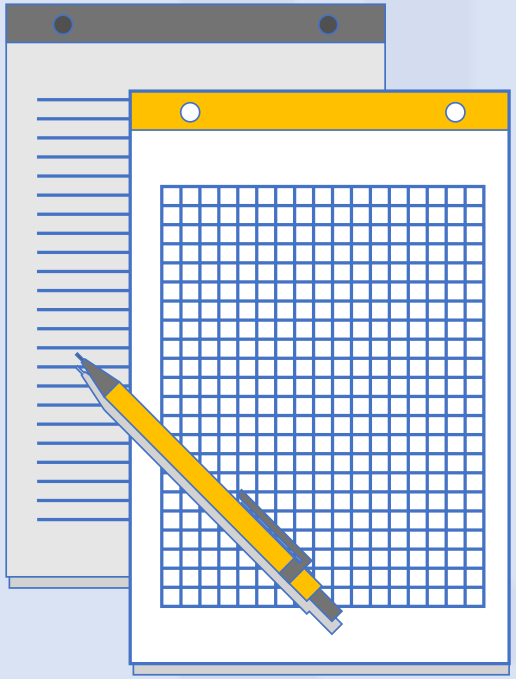
Accuracy

- Each model was analysed with the best parameter and accuracy then determined best_score

	Method	Score	Accuracy
0	Logistic Regression	0.833	0.846
1	Support Vector Machine	0.833	0.848
2	Decision Tree	0.833	0.875
3	K Nearest Neighbour	0.833	0.848

- Here, we can see from the 'accuracy' that though overall scores are the same, the **Decision Tree** model has slightly higher Accuracy rate, and is therefore the best model to use.

CONCLUSION : Findings & Implications



- Using SQL Visualisation we observe that the low flight numbers show lower success rate, whilst higher flight numbers show higher success rate.
- Similarly, the greater the mass of the payload, the higher the success rate.
- Using Mapping, we see that launch sites are near coasts and away from cities. They are also close to highways and rail for access.
- Using Dashboard observations we find that the success rate for low weighted payload is higher than heavy weighted payload.
- KSC LC-39A has the best record for the most successful launches from all sites

CONCLUSION : Findings & Implications

- When using Machine Learning algorithms and assessing the accuracy of scores, the **Decision Tree** Classifier Model is the best performer for predicting if the first stage of SpaceX rockets will land successfully.

- Considerations:
  - Dataset size (as small sample size was used for train- test of the models)
  - Alternative models, as scores were identical for all models, with only the best_score for accuracy showing marginal differences.

APPENDIX

```python
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```python
response = requests.get(spacex_url)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.047721
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.047721
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.047721
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9.047721
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857

```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

END

IBM Developer

SKILLS NETWORK